Modeling & Simulation of Distributed Systems

2015 Propulsion Control and Diagnostics Workshop
Cleveland, OH

Jonathan Kratz
with
Eliot Aretskin-Hariton, Dennis Culley, George Thomas and Alicia Zinnecker

Intelligent Control and Autonomy Branch
NASA Glenn Research Center

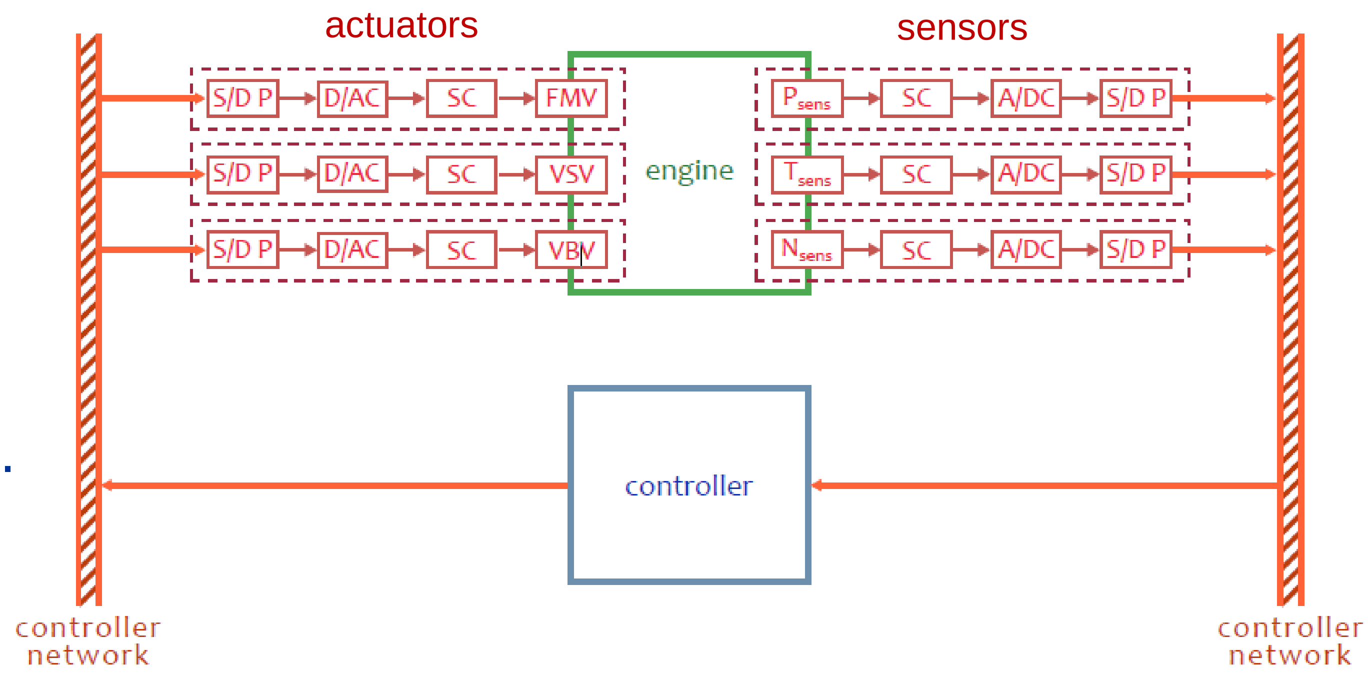
September 16, 2015

Outline

- Brief overview of the concept of distributed engine control
- Challenges for modeling distributed systems and creating a versatile hardware-in-the-loop (HIL) system
- Migration from a centralized to a distributed modeling approach
  - Decomposing an engine model
  - Modeling of control system components
  - Creating a library of re-usable modeling components
  - Establishing a template for modeling distributed systems
  - Working toward a hardware-in-the-loop (HIL) system
- Simulation Benchmarking and Comparison
- Real-time simulations with our Decentralized Engine Control Simulation System (DECSS)

Distributed Control

- Signal processing duties are moved to **smart transducers**
- **Digital data** is transferred between control components over a digital network.
  - Signal susceptibility to noise is reduced
  - Makes the control system more modular.
  - Off-loads some processing from the control unit
  - Network connecting the control components becomes important
    - Data loss, time delays, and data corruption

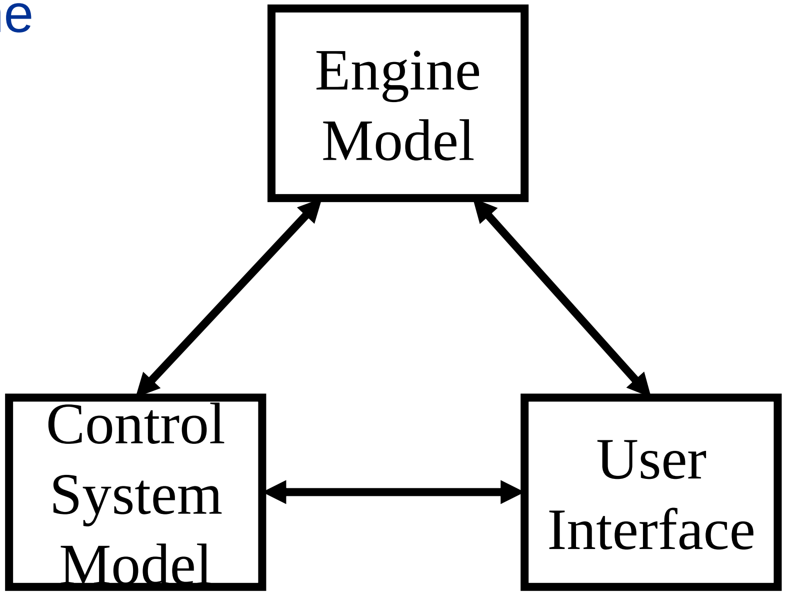


Working Toward a HIL System for DEC

- Challenges in modeling and simulating distributed systems
  - **Improved fidelity** of the control system
  - **Numerical precision** of the data used in simulations should reflect reality
  - **Reliable network models** are needed
  - Simulations should be able to mimic the **asynchronous nature** of an actual distributed control system with **different sampling periods**.
- Challenges in creating a versatile HIL system
  - **Proprietary models** and code must remain **protected**
  - Should **not be limited by their model development environment choice**
  - **Common interfaces**
  - Ability to dependably **run in real-time** and **interface with real hardware**.

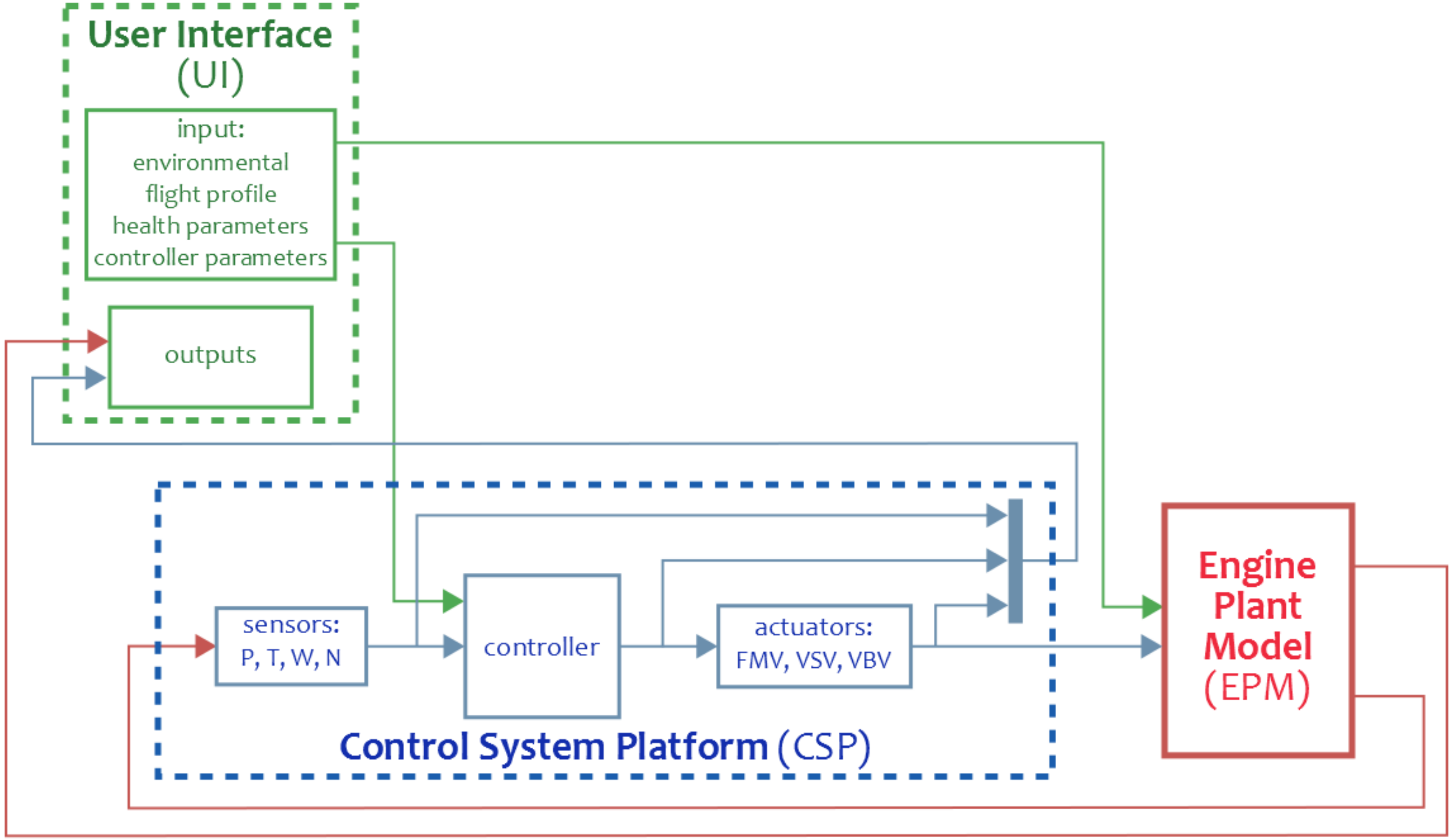
Migration from Centralized to Distributed Simulations

- The starting point was based around C-MAPSS40k
  - Serves as the engine model for demonstrating DEC
  - Use of the system is not limited to C-MAPSS40k or the MATLAB/Simulink environment it exists in.
- Unstructured Simulations (Breaking apart C-MAPSS40k)
  - Decompose the model and simulate each major component
    - Engine Model (EM)
    - Control Model (CM)
    - User-Interface (UI)
  - Each model is capable of being hosted on separate machines
  - A pre-defined set of data is transparently shared between the models



Migration from Central to Distributed Simulations

- Benefits of decomposing the model
  - Modularity – can easily replace one component with another
  - No specific software requirements
  - Proprietary models can be integrated in a simulation and remain protected
- Issues
  - Modularity adds some overhead that slightly increases execution time
  - Controlling asynchronous systems is not intuitively obvious

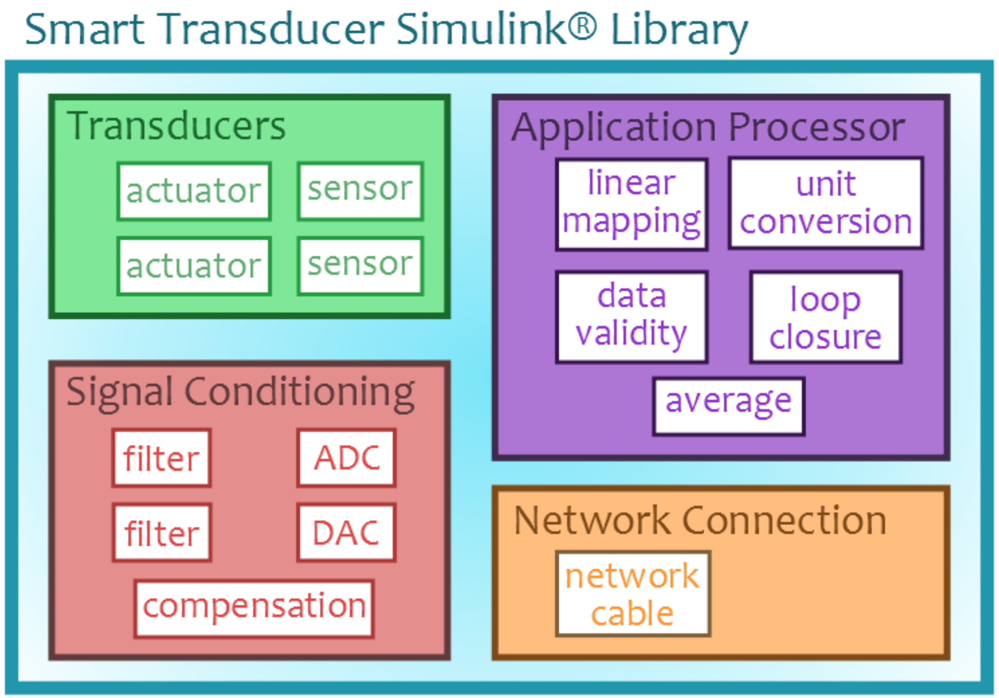


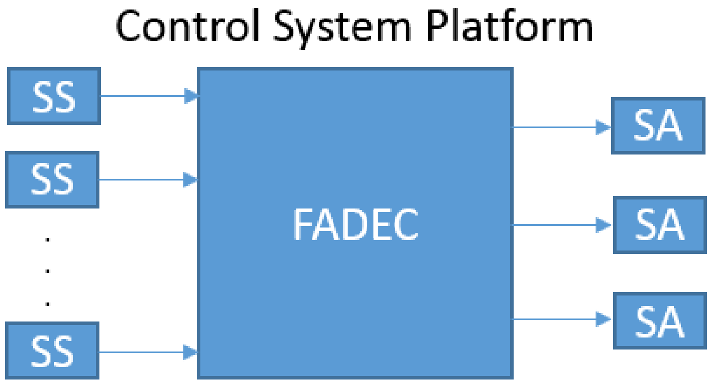
Control System Component Model Development

- Modeling changes occur in the control model which contains:
  - Sensor nodes
  - Actuator nodes
  - Controller
  - Controller network
- The basic functions of the control system were identified, modeled, and entered as library functions
- The library functions simplify the construction of more complex models and control architectures.
- Of special interest was the sensor and actuator models
  - Sensor & actuator models in C-MAPSS40k are first-order transfer functions – not sufficient for smart nodes in DEC applications

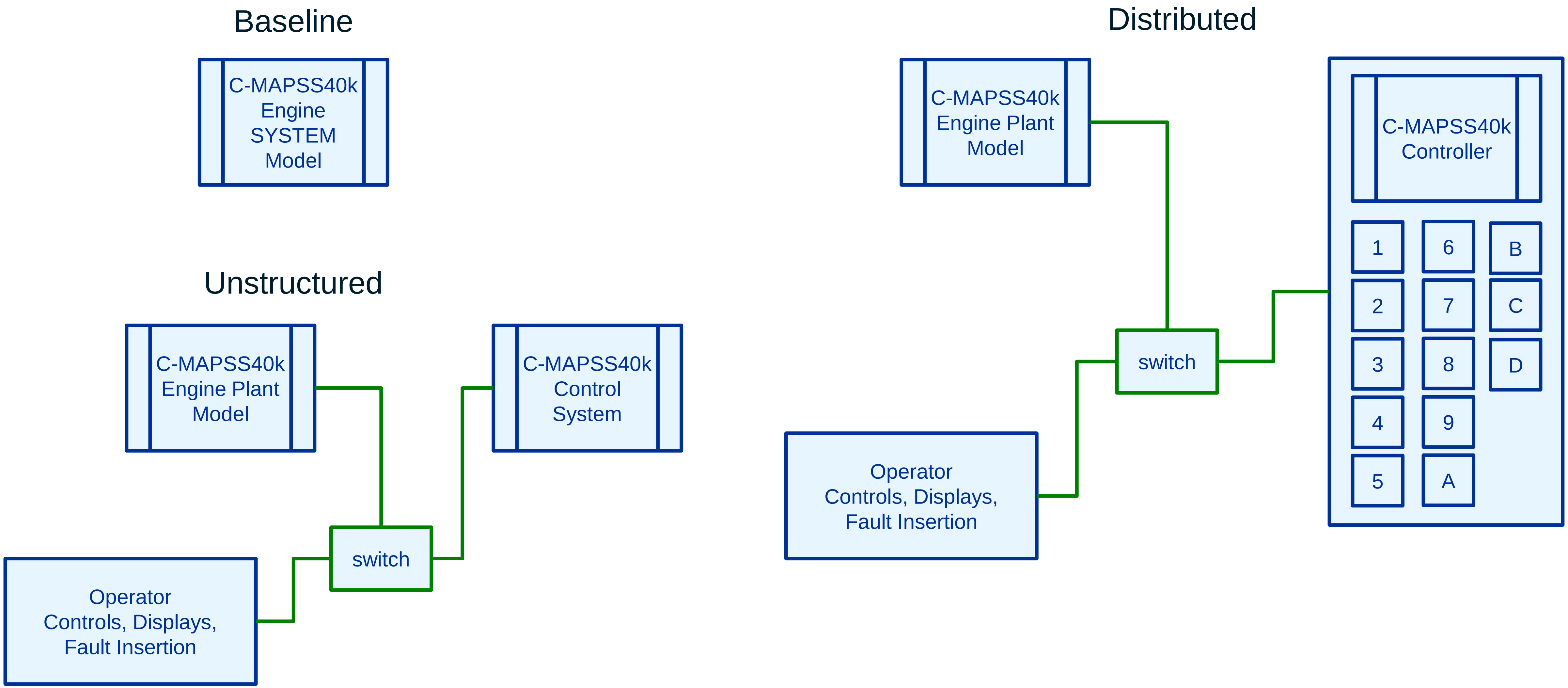
Control System Component Model Development

- How to model smart nodes?
  - IEEE 1451 specification with the following components:
    - Transducer hardware
    - Signal conditioning, conversion, and processing components
    - Network connection interface
- Used Simulink library to build-up smart node models
- Added smart sensor and actuator models into the simulation (distributed simulation)
- Included a simple network model
  - Randomly delays packets using a lognormal distribution to determine how much to delay the packet
  - Randomly drops a packet using a uniform distribution



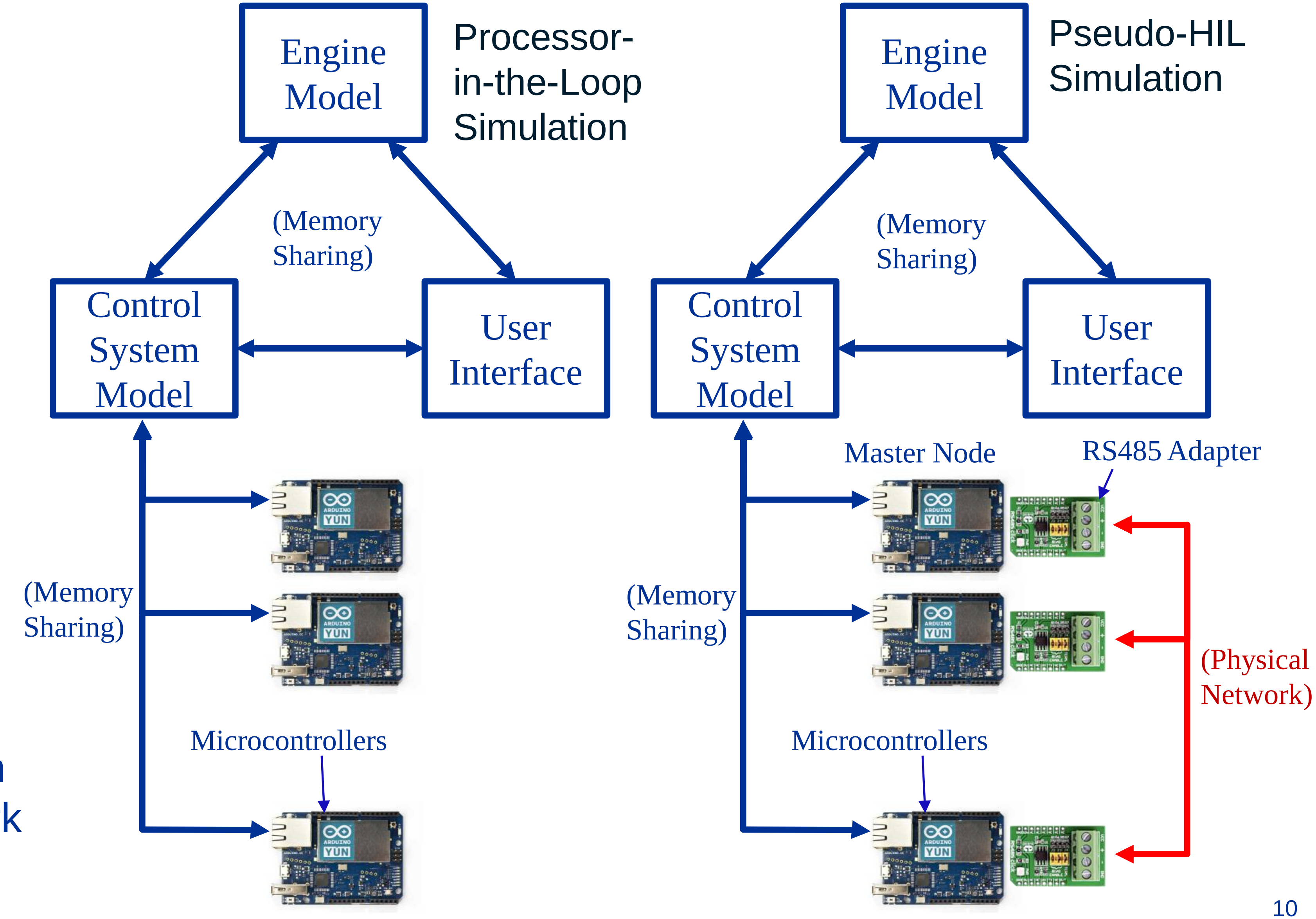


Simulation Progression Summary



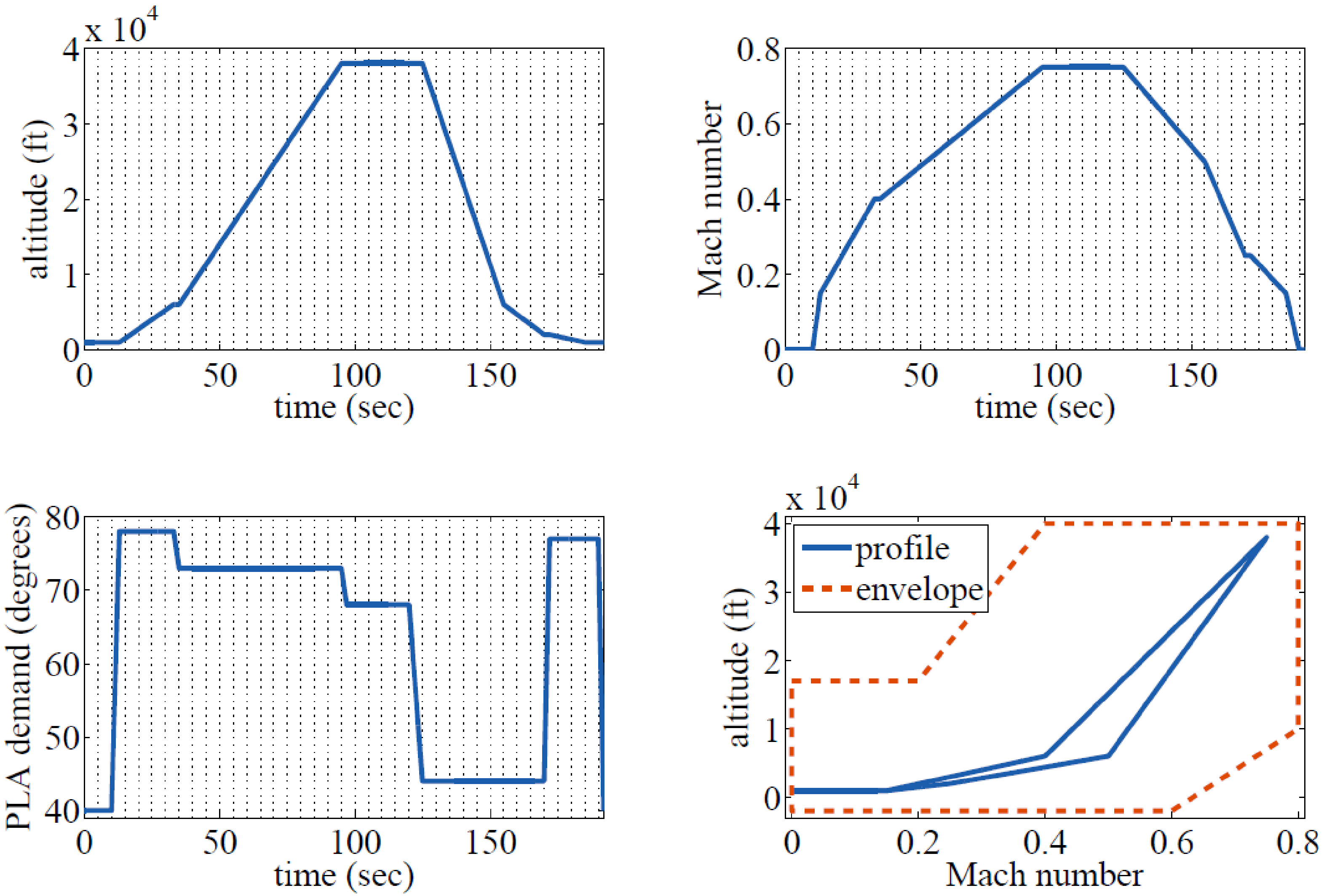
Microcontroller Extensions to the Simulation

- Processor-in-the-Loop
  - Smart nodes are simulated on their own dedicated microcontroller
  - Microcontrollers run on their own clock better illustrating the asynchronous nature of the control system
  - No physical network or network model implemented
- Pseudo-HIL
  - Brings a physical multidrop network into the loop
  - Simulation results may aid in the development of a network model

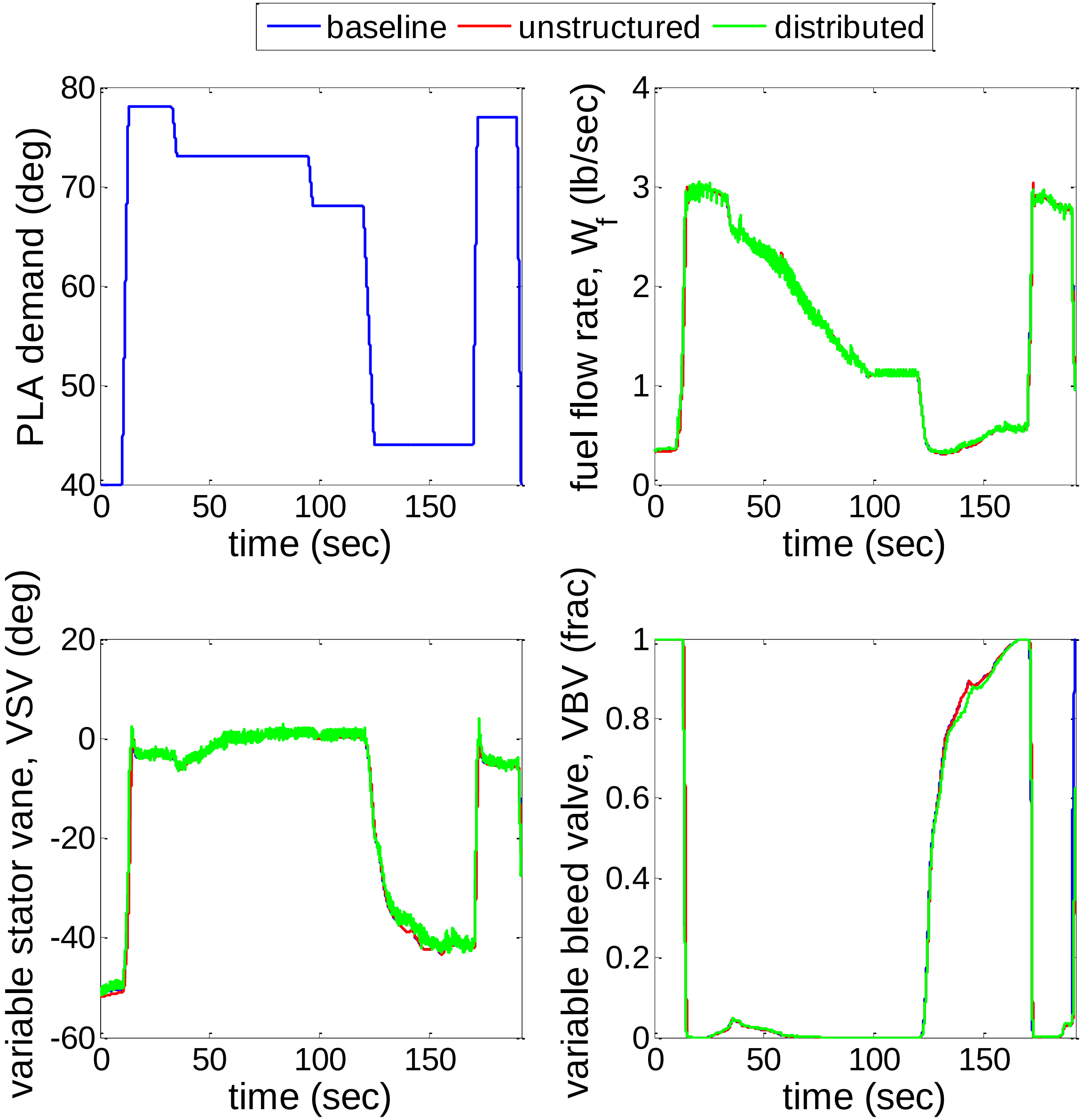


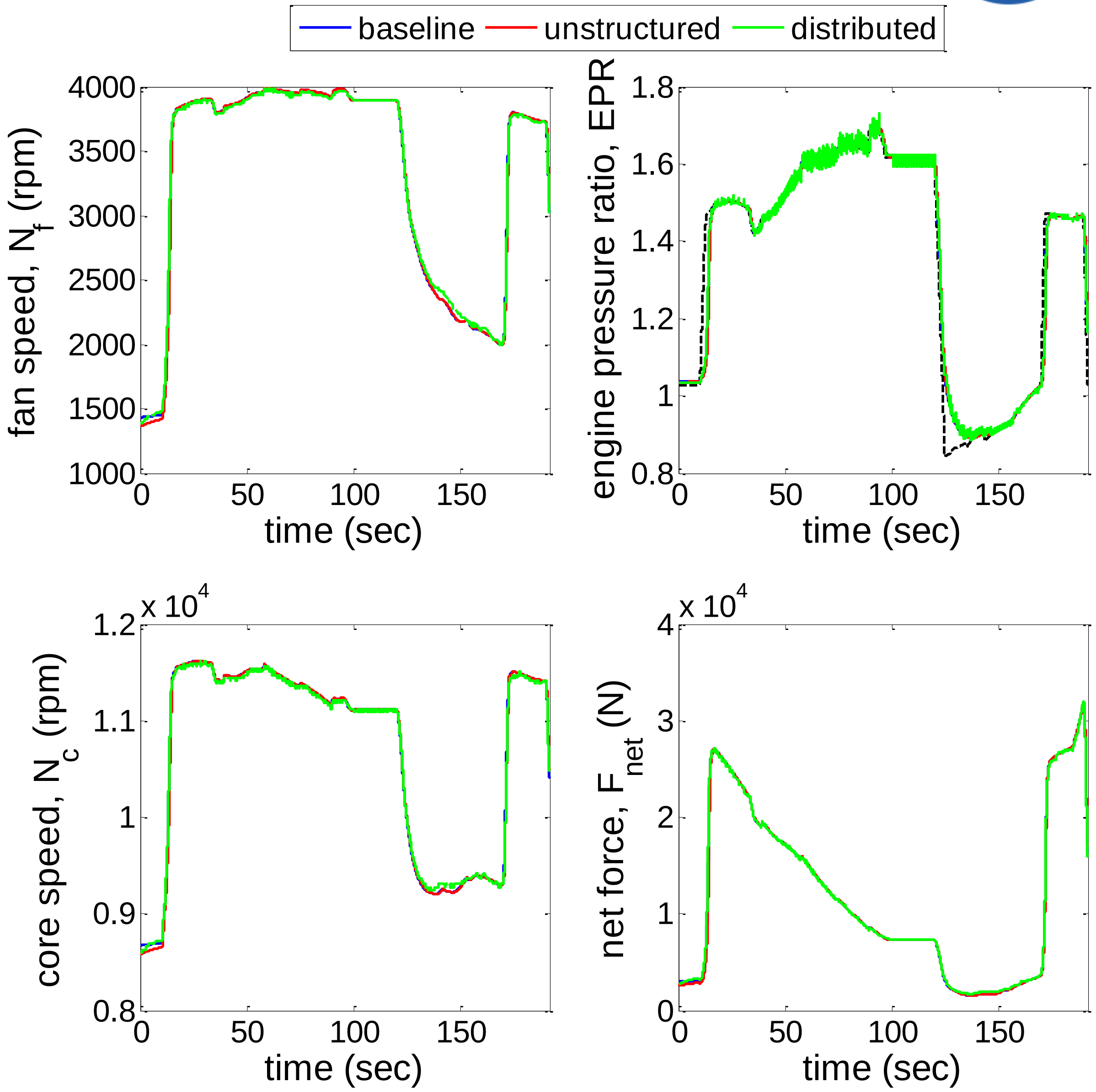
Simulation Benchmarking – Test Profile

- Each model described was simulated with the same flight profile

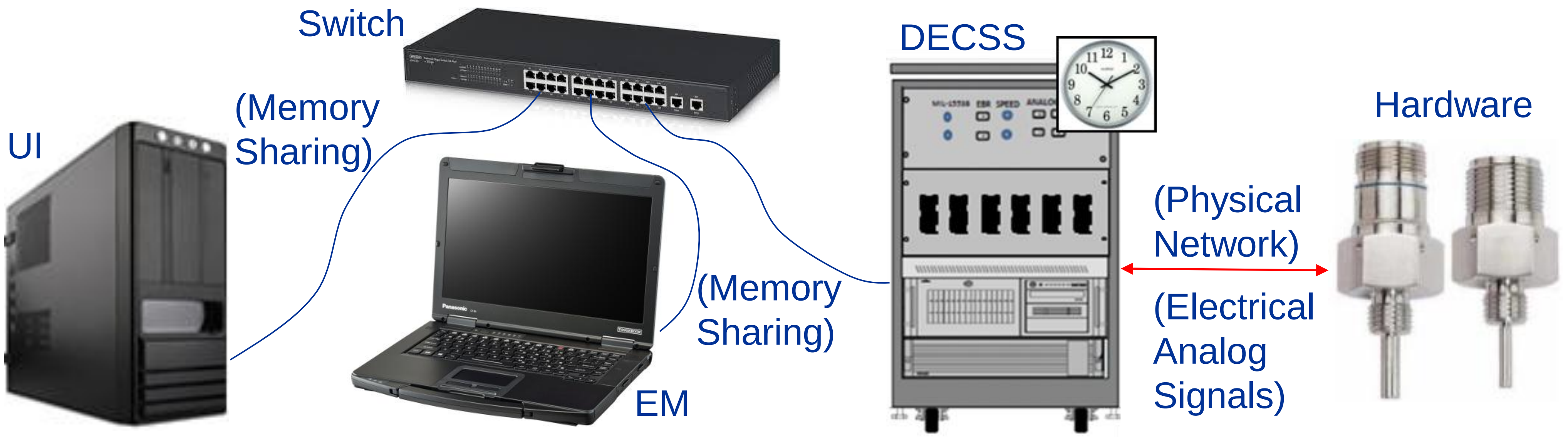


Simulation Benchmarking – Model Outputs



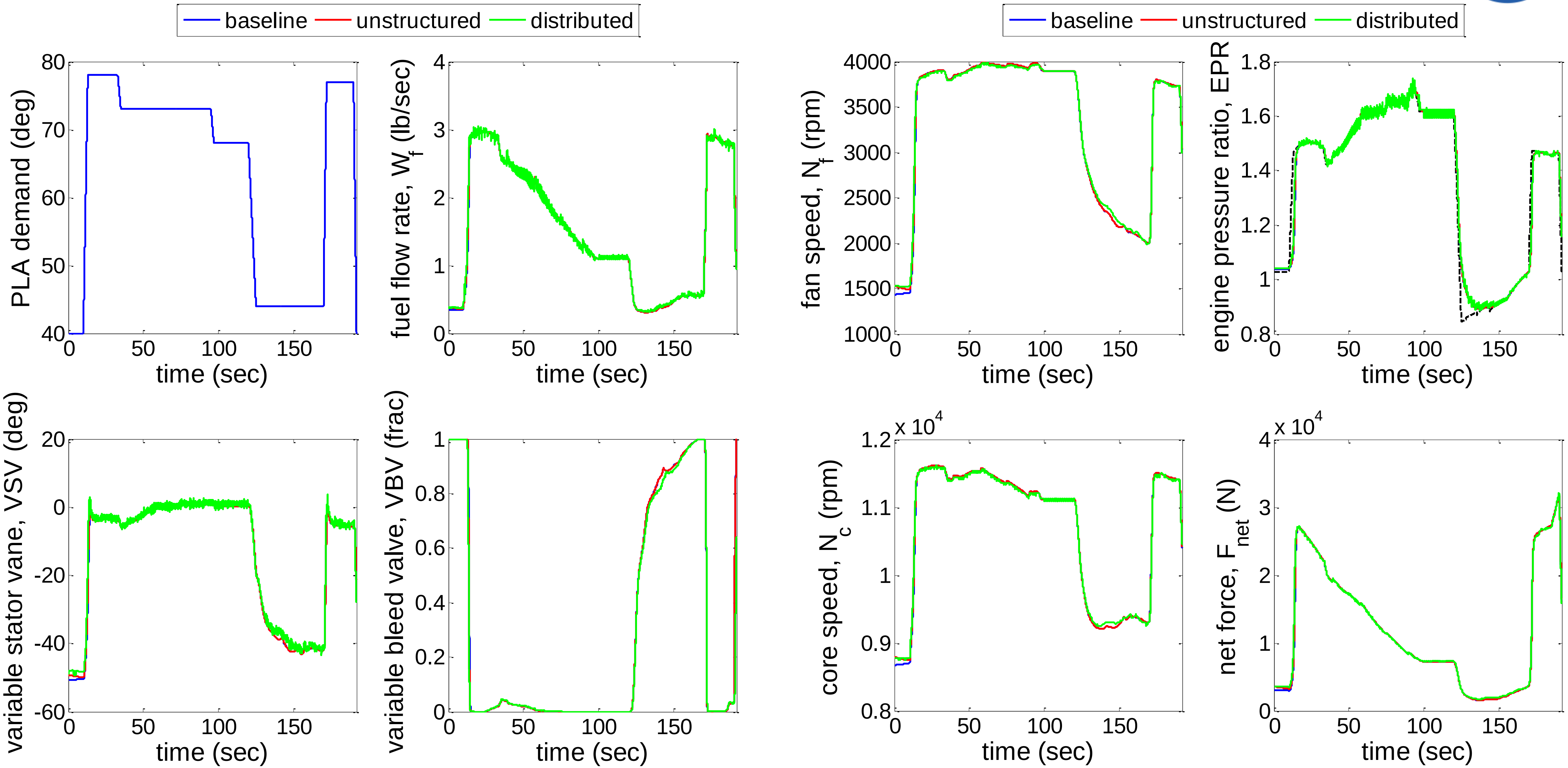


Real-Time Simulation with the DECSS



- DECSS hosts the control model (CM) in real-time simulations and perhaps other models as well
  - Each executable can have CPU, priority, and execution rate assigned
- SIMulation Workbench is used to setup, execute, and control the simulation as well as collect data
  - Real-time data analysis and plotting available
  - Can export data for analysis using another program

Real-Time Simulation with the DECSS



Taking Advantage of Multiple Frequency Based Schedulers

- Hardware sampling rates are not considered in the current control model
  - May operate asynchronously
  - May operate at different rates larger than the control interval
- DECSS has two, 8-core processors, each having multi-threading capabilities
- Each control element model becomes a process operating within the domain of a frequency based scheduler
  - 16 cores can be utilized to host the processes so that they emulate the asynchronous nature of a physically distributed control system
  - Processes do not need one common step-size (brute force) → run more efficiently
  - Gives complete control over the simulation
- SIMulation Workbench currently limits the user to using 1 frequency based scheduler
  - Efforts will be put toward resolving this issue

Hardware-In-the-Loop Capability

- Hardware-in-the-loop capability opens up collaboration opportunities
- DECSS can/will provide all electrical analog signals and control network communication interfaces to test hardware control elements.
- Plans are being made to use the DECSS functionality testing smart node hardware (Sporian Microsystems SBIR)
- Once functionality is demonstrated:
  - Studies can be conducted to
    - evaluate different control architectures and control networks
    - test control system hardware
    - develop and test new control algorithms for distributed systems
  - From these studies better models can be developed for the control system components and the controller network
    - Enable a faster and cheaper design process for smart nodes, control networks, and the overall distributed control system

Summary

- A structured methodology was followed to decompose the C-MAPSS40k engine system simulation into functional elements
- Libraries of these functional elements have been developed to create any engine system control architecture
- Several architectures have been created and validated against the original baseline engine system simulation
- Preliminary work has been started to investigate the asynchronous nature of distributed systems using microcontroller hardware
- These modeling techniques are now being applied to the DECSS which employs real-time parallel processing to simulate the asynchronous and multi-rate nature of distributed systems

References

- Culley, Dennis. ‘Transition in gas turbine control system architecture: Modular, distributed, and embedded’. In *ASME Turbo Expo2010: Power for Land, Sea, and Air*, volume 3, pp. 287Ű297, Glasgow, Scotland, United Kingdom, June 2010.\
- Culley, Dennis E., Thomas, Randy, and Saus, Joseph. ‘Concepts for distributed engine control’. In *Proceedings of the 43rd Joint Propulsion Conference and Exhibit*, No. AIAA-2007-5709, Cincinnati, OH, July 2007.
- Culley, D., Thomas, R., and Saus, J., ‘Integrated tools for future distributed engine control technologies,’ *Proceedings of the ASME Turbo Expo 2013*, No. GT2013-95118, San Antonio, TX, USA, June 2013.
- ‘IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats’, *IEEE Std 1451.0*, September 2007.
- ‘IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Network Capable Application Processor (NCAP) Information Model’, *IEEE Std 1451.1*, 2000.
- ‘IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Transducer to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats’, *IEEE Std 1451.2*, 1998.
- ‘IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Digital Communication and Transducer Electronic Data Sheet (TEDS) Formats for Distributed Multidrop Systems’, *IEEE Std 1451.3*, April 2004.
- Lee, Edward A.. ‘Modeling concurrent real-time processes using discrete events,’ *Annals of Software Engineering*, Vol. 7, No. 1-4, 1998, pp. 25-45.
- Lee, Kang. ‘IEEE 1451: A standard in support of smart transducer networking.’ In *Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference*, volume 2, pp. 525-528, Baltimore, MD, May 2000.
- May, R. D., Csank, J., Lavelle, T. M., Litt, J. S., and Guo, T.-H., ‘A High-Fidelity Simulation of a Generic Commercial Aircraft Engine and Controller,’ *Proceedings of the 46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, No. AIAA-2010-6630, Nashville, TN, July 2010.
- Song, Eugene Y. and Lee, Kang. ‘Understanding IEEE 1451 - Networked smart transducer interface standard,’ *IEEE Instrumentation & Measurement Magazine*, April 2008, pp. 11-17.
- Zinnecker, A., Culley D., Aretskin-Hariton, E., ‘A Modular Framework for Modeling Hardware Elements in Distributed Engine Control Systems’. In *Proceedings of the 50th Joint Propulsion Conference and Exhibit*, No. AIAA-2014-3530, Cleveland, OH, July 2014
- Zinnecker, A., Aretskin-Hariton, E., Culley D., ‘Benchmarking variants of a hardware-in-the-loop simulation system’. Extended abstract for the *AIAA Science and Technology Forum and Exposition (SciTech 2016)*, San Diego, CA, January 2016.

Questions/Discussion

jonathan.kratz@nasa.gov